



INFORME DE LABORATORIO

Autores: *Angee Lorena Ocampo Ramírez, Oscar Andrés Gutiérrez Rivadeneira*

*Laboratorio de Electrónica Digital 2
Departamento de Ingeniería Electrónica y de Telecomunicaciones
Universidad de Antioquia*

Resumen

En la presente práctica se desarrolla un multiplicador de números de punto flotante empleando el formato estándar IEEE-754, este multiplicador realiza dicha operación entre los números contenidos en el rango, además, realiza ciertas operaciones con números extremos, los cuales fueron denominados como casos especiales; esta práctica tomó como base la teoría proporcionada durante las sesiones de clase, posteriormente, se implementó el diseño mediante el lenguaje de descripción de hardware SystemVerilog, el cual posee la herramienta de simulación ModelSim, para evaluar el diseño creado, y, finalmente, visualizar su comportamiento mediante la FPGA.

Palabras clave: Multiplicador, Punto Flotante, FPGA, IEEE-754, System Verilog

Procedimiento

Ahora es posible observar el análisis efectuado para la elaboración de la práctica, igualmente, se muestra el diseño de la misma y su respectiva simulación. Para el desarrollo de esta práctica, se partió de la plantilla dada por el docente, en la cual se especificaban una estructura que podía resultar útil para realizar la práctica, además de indicar los módulos necesarios para capturar, procesar y mostrar los valores deseados.

Diseño:

Como se mencionó anteriormente, la plantilla proporcionada estaba compuesta por 8 módulos, incluyendo la

entidad principal.

El primero de estos fue *peripherals*, este módulo se encarga de generar las señales y manejar los periféricos que indican en qué momento el multiplicador se encuentra cargando los datos y cuándo está generando los resultados, esto lo hace mediante dos contadores y señales que reciben los respectivos pulsos ingresados a través de la tarjeta, uno de los contadores es usado para cargar los datos y otro para la salida de resultados; además este módulo, es el responsable de instanciar el *peripheral_getoperands* que se ocupa de cargar la data de ingresada en un arreglo de 64 bits con el propósito de asignar los datos a cada una de las entradas, este proceso se realiza mientras *loaddata* se encuentre en 1; por otro lado, *peripherals* también se ocupa de instanciar la entidad *peripheral_deco7seg*, esta última hace referencia a los decodificadores que serán usados para mostrar las letras y el respectivo resultado de acuerdo a cada una de las señales y banderas. Es importante resaltar que esta entidad también contiene el módulo que produce los pulsos cada vez que el botón es presionado.

Seguidamente, la plantilla contenía la unidad de datapath (*datapathunit*), esta entidad se responsabiliza de hacer las respectivas conexiones entre los periféricos de la tarjeta, los cuales fueron descritos anteriormente, y el multiplicador (*multiplierunit*); este multiplicador maneja el formato estándar IEEE 754, debido a esto dentro de él se segmentan las entradas para obtener el exponente, signo, y fracción de cada una de ellas; posteriormente, se procede a concatenar un 1 a cada una de las entradas para normalizar las mantisas, después de esto se ejecuta la multiplicación de estas, y una XOR entre signos para conseguir el respectivo signo del resultado; luego, antes de generar el exponente del mismo, primero se hace una evaluación de los casos especiales, debido a que si esta verificación no es realizada el resultado producido será erróneo, dado el caso de que se presente alguno de los ca-

son considerados como especiales (0,-0, ∞ , $-\infty$ y NAN), se indica mediante un arreglo de banderas que especifica cuál es el caso que está sucediendo; si los números ingresados se encuentran dentro del rango, el multiplicador determina si es necesario una normalización de los exponentes, para, finalmente ensamblar el resultado acorde al formato seguido.

Asimismo, se incluye la unidad de control, *controlunit*, esta unidad es la que crea la máquina de estados requerida para cuando se estén cargando los datos de entrada, y cuando este proceso finaliza, comience a visualizarse el resultado en la tarjeta, por lo anterior, aquí se actualiza la señal de *loaddata* que depende a su vez de la entrada *inputdata_ready*, esta entrada es la responsable de indicar cuando los datos de entrada se encuentran listos para pasar al próximo estado. Por último, está el módulo *top*, en este se crean las respectivas señales y conexiones para instanciar las unidades de control y *datapath*.

Simulación

Para esta práctica se realizaron 4 testbench diferentes para poder conocer el comportamiento de los módulos más importantes, estos eran el *datapath*, multiplicador, unidad de control y finalmente la que los contiene a todos, el *TOP*. Para el desarrollo de los mismos se tuvo en cuenta la jerarquía que tenían dichos módulos, inicialmente se realizaron las pruebas en el multiplicador para verificar que realizara las operaciones con éxito, además de poder cumplir con los casos especiales; luego, las pruebas se realizaron en el *datapath*, que permitió observar si las conexiones entre los periféricos y el multiplicador estaban funcionando de manera correcta. Siguiendo con las pruebas se hizo el testbench para la unidad de control, el cual fue muy sencillo al tratarse de una máquina de dos estados; no obstante, era vital verificar el cambio en las banderas para definir cuando recibir datos y cuando no; por último, se realizaron pruebas en el *TOP*, módulo que contiene todos los anteriores, permitiendo verificar el comportamiento completo, y que todos los módulos operaran de la manera esperada.

Para las pruebas del multiplicador se ingresaron valores manualmente, los cuales permitiesen verificar el funcionamiento de la bandera de casos especiales, en este punto el resultado obtenido no es relevante, ya que dado un caso especial los displays 7 segmentos ignoran el *dataR* y simplemente indican el caso. Se culminan las pruebas con valores normalizados y el resultado es verificado con [?]. El resultado de la simulación se puede

apreciar en la imagen **0-1**, donde principalmente se pueden analizar los datos de entrada, la salida o resultado y la bandera para casos especiales.

Una vez se comprobó que el multiplicador funcionaba de manera correcta, se realizaron pruebas en el *datapath* (Culminadas las pruebas realizadas al multiplicador, se procede a comprobar el funcionamiento del *datapath*), en este punto el objetivo era el de comprobar todas las conexiones realizadas, ya que un error en este punto podría significar un funcionamiento incorrecto, además de verificar si la señal *inputdata_ready* se activase de manera correcta, además de comprobar las señales *dataA* y *dataB* se almacenen de forma correcta, tal como se puede apreciar en la figura **0-2**.

El testbench más sencillo de realizar fue el de la unidad de control, ya que solo se debía comprobar que la máquina de estados hiciese el cambio cuando recibe la señal *inputdata_ready* y que el *loaddata* sea cero, además de que si se reiniciara con él *reset*, justo cómo ocurre en la figura **0-3**.

Finalmente, se realizó el testbench para el módulo llamado *top*, el cual integra el *datapath* y la unidad de control, por lo que solo se quería comprobar que todas las conexiones estuviesen bien, ya que anteriormente se pudo comprobar que los módulos funcionaban correctamente. Este testbench se puede apreciar en la figura **0-4**.

Conclusiones

Durante el desarrollo de esta práctica se pudo ver la importancia de la creación de los módulos que posibilitaron segmentar el problema propuesto, además, el uso de los test bench de manera estratégica, sirvió para evaluar el comportamiento del diseño requerido. Por otro lado, también se apreció la complejidad que poseen este tipo de operaciones a nivel del hardware, debido a todas las evaluaciones y conexiones necesarias para su correcto funcionamiento. Es importante resaltar, que comenzar el proceso de diseño analizando el problema antes de proceder a plantearlo mediante el programa de descripción de hardware, es un paso fundamental para simplificar un poco el ejercicio propuesto; el tiempo destinado para el desarrollo y culminación de este dilema, fue de aproximadamente dos horas y media por día durante una semana, teniendo en cuenta que los integrantes del equipo presentaron ciertas confusiones que retrasaron un poco su entrega.

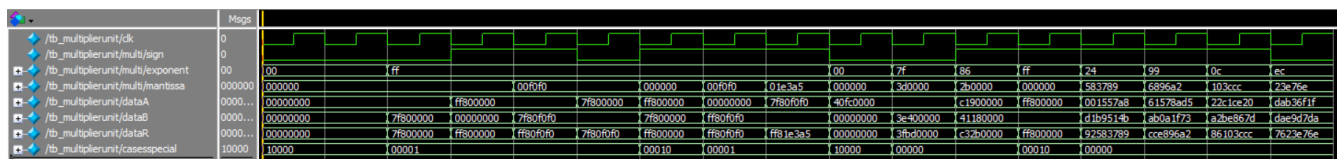


Figura 0-1: Testbench Multiplicador.

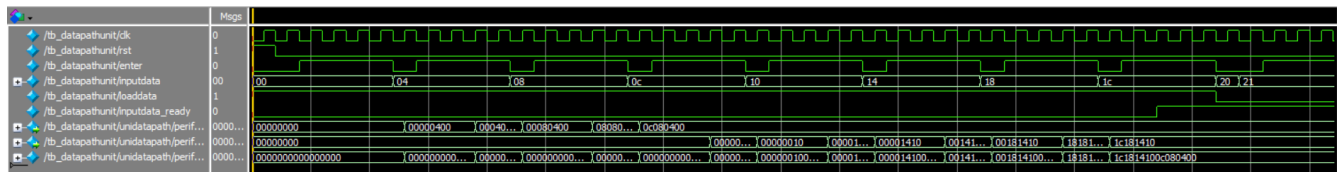


Figura 0-2: Testbench Datapath.

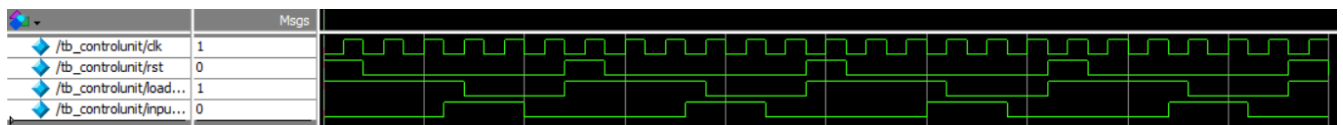


Figura 0-3: Testbench unidad de control.

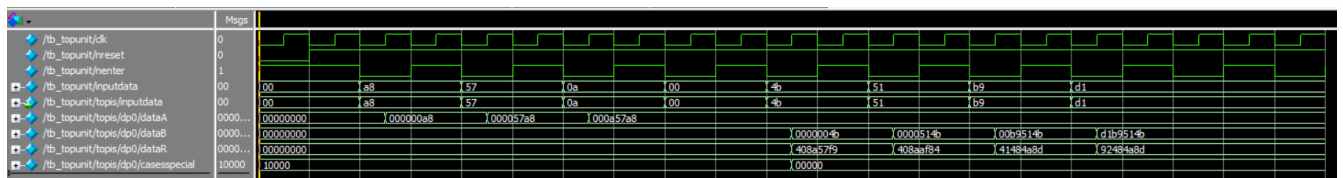


Figura 0-4: Testbench Datapath

Bibliografía

- Edmund Weitz. IEEE 754 Calculator. URL <http://weitz.de/ieee/>.